KT Final Reviewing!!! Fighting!!!

Web Search

Chapter1 Introduction

1. Concrete Tasks: The task is well-defined, can assess whether the solution is correct.
2. Problem Categories: Simple/Complicated/Complex/Chaotic/Disorder
3. Knowledge Tasks:
   - Task where the data is irregular and unreliable, or the outcome is not well-defined.
   - Not a computational task, but in such tasks, we do use computers to mediate us and data, in helping to reach a decision
4. Characteristics of knowledge technologies:
   - general (e.g. machine learning): the data must be transformed to suit the axioms or assumptions of the method, in a rigorous way
   - specific (e.g. machine translation): detailed understanding of the task is used to drive development of the method
   - Example:
     A specialized problem: parse Chinese to a particular language
     A approximate problem: parse Chinese to a language
     A general problem: parse a language to another language

Chapter2 String Search

1. Regular expression: are patterns that match character strings.
   - Search: Find the strings in a file that contain a substring that matches a given pattern (egrep)
   - Find and replace: Substitute some new string for the matching substring (s/pattern/replace/g)
   - Validate or test: Check if new string is correct (input ~/string/ 在输入的子串中查找是否有 string 这个字符串)
   - ---
   - Metacharacters: { } [ ] ( ) ^ $. | * + ? \ , they need to be escaped by a backslash (\) to be used in a literal match
   - Matching：
     - The foundation of regex is literal matching.
     - Matches are case sensitive
     - Whitespace is significant
     - Substrings are uninterpreted (/lane/ will match “planet”)
     - The wildcard . is the most basic metacharacter (通配符)
     - The anchors ^ and $ match the start and end of a line or string (防止匹配子串)
   - Alternation:
     - The | metacharacter expresses alternation or disjunction
     - the | character has low precedence (优先级)
   - Repetition：
     - The precise number of characters to match may be unknown, so

specify a repetition construction. (*,+,?)

  - ➢ Greedy: match as many characters as they can (e.g. a.*b will pick up the last "b" in the string.)

- Character classes
  - ➢ Match a set of characters
  - ➢ Example: [Kk],[aeiou] = /a|e|i|o|u/, [A-Z]
- Negative classes
  - ➢ A second use of ^ metacharacter is to negate character classes. (E.g. /[^A-Za-z]/ matches ant non-alpha character)
- Named classes
  - ➢ Some character classes are used so frequently that they have names
  - ➢ [0-9] = [[:digit:]] = \d
  - ➢ [a-zA-Z0-9_] = [[:word:]] = \w
  - ➢ [\ \t\r\n\f] = [[:space:]] = \s
  - ➢ As do their negations: \D,\W,\S
- Back-reference or memorization
  - ➢ Placing a pattern in parentheses leads to the match being stored as a variable. (\1....\n)

Chapter3 Approximate String Search and Matching

1. Two main applications for Approximate String Search:
   - Spelling correction (need the notion of dictionary)

     Problems: (出没出现都有可能是拼写错误)

     1) Item in the input which doesn't appear in the dictionary is misspelled.

     2) Item in the input which does appear in the dictionary might be correctly spelled or misspelled (out of scope)
   - Computational Genomics

2. Neighborhood Search
   - Algorithm:
     - ➢ For a given string w of interest:
     - ➢ Generate all variants of w that utilize at most k changes (Insertions/Deletions/Replacements) --- neighbors (小于 k 次修改的结果也会返回)
     - ➢ Check whether generated variants exist in dictionary
     - ➢ All results found in dictionary are returned
   - Efficiency Analysis: surprisingly fast

3. Edit Distance

   1) Global Edit Distance:
   - Algorithm:
     - ➢ Transform the string of interest into each dictionary entry, using the operation insert, Delete, Replace and Match
     - ➢ Each operation is associated with a score
     - ➢ Best match is the dictionary entry with best aggregate score (score 的大小由 Match 指令决定)
   - Levenshtein Distance (true distance): number of edits required to

transform one string into the other (symmetric) Match(0), Insert(+1), Delete(+1), Replace(+1)

- Needleman-Wunsch algorithm (Match score greater than others-→max())

2) Local Edit Distance

- Algorithm:
  - Like GED, but searching for substring match.
  - Particular suitable when comparing two strings of vary different lengths (e.g. a word and a sentence) (适用于两个串长度有很大差别的情况)
- Smith-Waterman algorithm (Match mush have different +/- sign to others)

3) Efficiency Analysis:

- Both algorithms time and space complexity is O(|f||t|)
- Integer comparisons are roughly the number of characters in the dictionary. Whether this is feasible depends on the size of the dictionary. (算法是否可行由字典大小决定)

4. N-Gram Distance (true "distance")

- Algorithm:
  - Compare two strings to determine "best" match
  - (character) n-gram: substring of length n
  - N-Gram Distance between n-grams of string s and t:
    |Gn(s)| + |Gn(t)| - 2*|Gn(s)∩Gn(t)|
- Efficiency analysis:
  - More sensitive to long substring matches, less sensitive to relative ordering of strings (matches can be anywhere)
  - Quite useless for very long string and/or very small alphabets

5. Suitability for Computational Genomics

- N-Gram can (almost) work! Tends to prefer shorter chromosomes like GED.

6. Soundex

- Algorithm:
  - Except for initial character, translate string characters according to table
  - Remove duplicates (e.g. 4444->4)
  - Remove 0s
  - Truncate to four symbols

7. Other phonetic Method

- Editex: uses the Edit Distance to compare strings based on a similar translation table to Soundex
- Ipadist: uses a text-to-sound algorithm to represent tokens according to the International Phonetic Alphabet

8. Evaluation: consider whether the system is effective at solving the user's problem.
   - Accuracy: fraction of correct response
   - Precision: fraction of correct response among attempted response
   - Recall: proportion of words with a correct response

Chapter4 Text Search

1. Information Retrieval (IR)
   - Definition: the subfield of computer science that deals with storage and retrieval of documents
   - What distinguishes IP from other fields? (other fields(DB, file structure) deal with storage, retrieval and also computation using the retrieved data in general)
     - ➢ There is an emphasis on the user. IR systems can be characterized as mechanism for finding documents that are of value to an individual.
     - ➢ The meaning or content of a document is of more interest than the specific words used to express the meaning.
   - Documents aren't always text. They can be defined as messages, an object that conveys information from one person to another.
   - Data retrieval vs. IR:

     DR:
     - ➢ Conventional DB systems, such as relational systems, are designed for data retrieval.
     - ➢ The information is unambiguous.
     - ➢ Queries are represented in an algebraic(代数的) language

     IR:
     - ➢ The stored documents are real-word objects that have been created for individual reasons. They do not have to have consistent format.
     - ➢ Documents are rich and ambiguous.
     - ➢ There is no conceivable (可能的) automatic method for translating then into algebraic form.

     - ➢ DR is used to retrieve items based on facts that describe them. (标题)
     - ➢ IR is used to retrieve items based on their meaning. (内容)

2. Information seeking
   - Information needs: (因为有 IN 所以有 IR)
     - ➢ Different kinds of IR system are linked by the concept of IN
     - ➢ An IR system is used by someone because they have an IN they wish to resolve.
     - ➢ Many IN cannot be described succinctly (算法的) (没有一种算法的解决方式)
   - Answers:
     - ➢ An answer to a query could be defined as a document that matches the query according to formal criteria. (e.g. if it contains all the query

words)

- Document should be relevant. (The relevance of a document to an IN cannot be determined computationally.)

  Relevance Definition: a document is relevant if it contains knowledge that helps the user to resolve IN.

3. Approaches for retrieval: consider the criteria that a human might use to judge whether a document should be returned in response to a query. (There is not computational way of approximating this process. Instead, we have to develop methods that use other forms of evidence to make a guess as to whether a document is relevant.)

- Boolean querying:
  - A typical query : diabetes & risk & factor & NOT juvenile
  - Documents match if they contain the terms, and don't contain the NOT terms.
  - There is not grey: matching is yes/no
  - 算法可以更复杂一点---- 近似匹配
  - --
  - It is repeatable, auditable and controllable.
  - The time in developing precise queries is a bit long (months)
  - It is unsatisfactory in several respect: no ranking/no control over result set size/ difficult to incorporate factors such as monotonicity (单调性)/remarkable to do well.

- Ranking:
  - A query is matched to a document by looking for evidence in the document that it is on the same topic as the query.
  - The more similar or likely a page is, relative to the other documents in the collection, the higher its rank. (相似度越高,rank 值越高)
  - Similarity:

    Use synonyms or related terms (利用近义词):

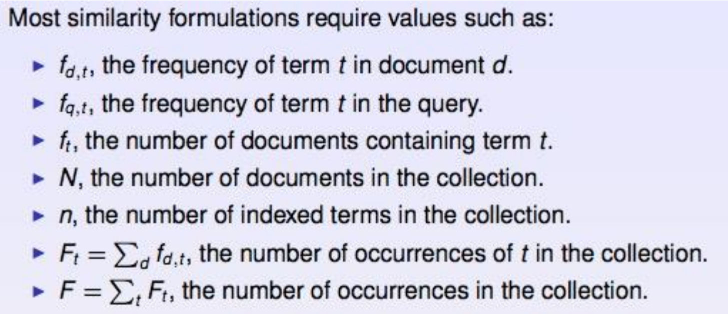
    1) Query expansion: fetch similar documents, find the terms have these in common and add these to the query.
    2) Relevance feedback: ask the user which documents were right, and then proceed as for query expansion.
  - Monotonicity:
    1) Less weight is given to terms that appear in many documents (Inverse document frequency or IDF)
    2) More weight is given to terms that appear many times in a document. (In-document frequency or TF)
    3) Less weight is given to documents that have many terms.

       The intention is to bias the score towards relevant documents by favouring terms that seem to be discriminatory (有辨识度的) and reducing the impact of terms that seem to be randomly distributed.

       (前两点是对于 terms 而言，最后一点对于 document 而言)

4. Principles & models

- The vector-space model:
  - ➢ Suppose there are n distinct indexed terms in the collection. Then each document d can be thought of as a vector
    <Wd,1, Wd,2 ….. Wd.n>
    where Wd,t is a weight describing the importance of term in d. (term 在指定文件中的权值)
    如果两个向量<Wd,1, Wd,2 ….. Wd.n> & <Wd’,1, Wd’,2 ….. Wd’.n>相似，说明 document d 和 d’相似
  - ➢ The space is orthogonal (Cartesian) – that is, the terms are treated as if they occur independently (虽然这个假设是错的)
  - ➢ Most similarity formulations require values such as:



Most similarity formulations require values such as:

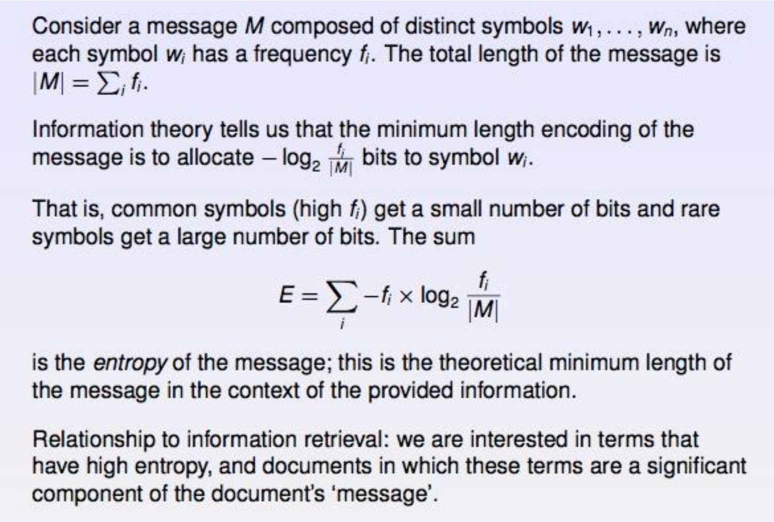
- $f_{d,t}$, the frequency of term t in document d.
- $f_{q,t}$, the frequency of term t in the query.
- f_t, the number of documents containing term t.
- N, the number of documents in the collection.
- n, the number of indexed terms in the collection.
- $F_t = \sum_d f_{d,t}$, the number of occurrences of t in the collection.
- $F = \sum_t F_t$, the number of occurrences in the collection.

To link back to monotonicity: we wish to find documents d that have

1) Term t with low ft, that is, are rare (较少 document 出现这个词)
2) but t has high fd,t, that is common in the document (在 document 中的词频高)
3) And $\sum_{w \in d} f_{d,w}$ is low, that is, the document is short.

- The cosine measure
  - ➢ Measure the angle between vectors

- Information theory



Consider a message M composed of distinct symbols $w_1, \ldots, w_n$, where each symbol w_i has a frequency f_i. The total length of the message is $|M| = \sum_i f_i$.

Information theory tells us that the minimum length encoding of the message is to allocate $-\log_2 \frac{f_i}{|M|}$ bits to symbol w_i.

That is, common symbols (high f_i) get a small number of bits and rare symbols get a large number of bits. The sum

$$E = \sum_i -f_i \times \log_2 \frac{f_i}{|M|}$$

is the *entropy* of the message; this is the theoretical minimum length of the message in the context of the provided information.

Relationship to information retrieval: we are interested in terms that have high entropy, and documents in which these terms are a significant component of the document’s ‘message’.

熵是理论上 message 的最短编码长度．熵越大越好，说明越多词频少的

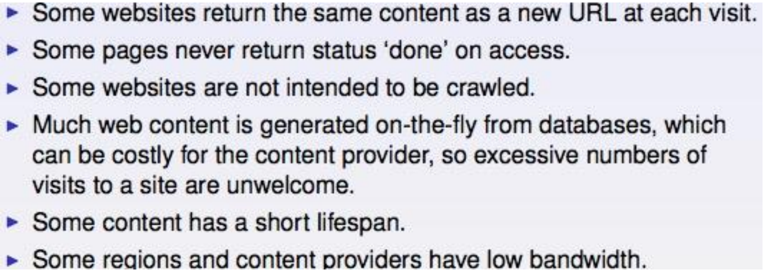
keyword 被算进去

Chapter5 Web Search

1. Element of a web search engine
   - Web search involves four main technological components:
     - Crawling: the data to be searched needs to be gathered from the web.
     - Parsing: the data then needs to be translated into a canonical (典范的) form.
     - Indexing: data structures must be built to allow search to take place efficiently
     - Querying: the data structure must be processed in response to queries

2. Crawling:

   Before a document can be queried, the search engine must known that it exists. On the web, this is achieved by crawling.
   - Basic challenge: there is no central index of URLs of interest.
   - Secondary challenge:



- Some websites return the same content as a new URL at each visit.
- Some pages never return status ‘done’ on access.
- Some websites are not intended to be crawled.
- Much web content is generated on-the-fly from databases, which can be costly for the content provider, so excessive numbers of visits to a site are unwelcome.
- Some content has a short lifespan.
- Some regions and content providers have low bandwidth.

3. Page Recognition

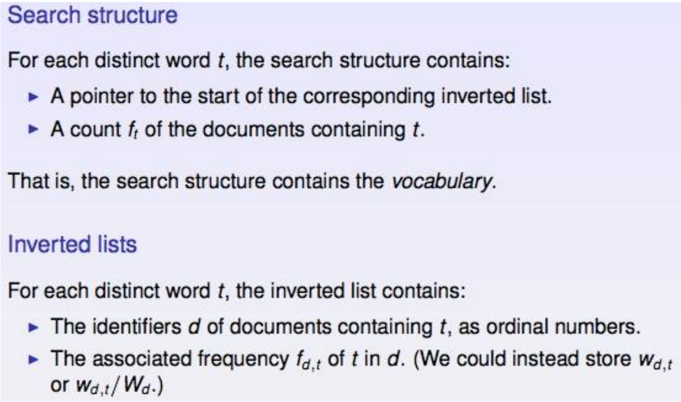
   Once a document has been fetched, it must be parsed. That is, the words in the document are extracted, and then added to a data structure that records which documents contain which words.
   - Page analysis
     - Tokenisation: The aim of parsing is to reduce a web page, or a query, to a sequence of tokens. (将大段文字解析成一个个 token)
     - Canonicalisation (正则化)
       1) Discard stop words (e.g. the, or…)
       2) Discard terms that linguistic rules suggested were not reasonable query strings (word 的长度不超过 64 个 characters)
       - ✓ Stemming: is the process of stripping away affixes(词缀)

         E.g. ies-→I, ational-→ at…..
       - ✓ Zoning: Web documents can usually be segmented into discrete zones such as title, anchor text, heading, and so on. Parsers also consider issues such as font size, to determine which text is most prominent on the page and thus generate further zones. Web search engines typically calculate weights for each of these zones and compute similarities for documents by combining these results.

4. Indexing:

Fast query evaluation makes use of an index. A data structure that maps terms to the documents that contains them.

- Inverted file: a collection of lists, one per term, recording the identifiers of the documents containing that term. (每个 term 都有一个 list, 包含有哪个 document 包含这个 term (可以包含 term 的位置以及词频数))
- Inverted file components:



Search structure

For each distinct word t, the search structure contains:

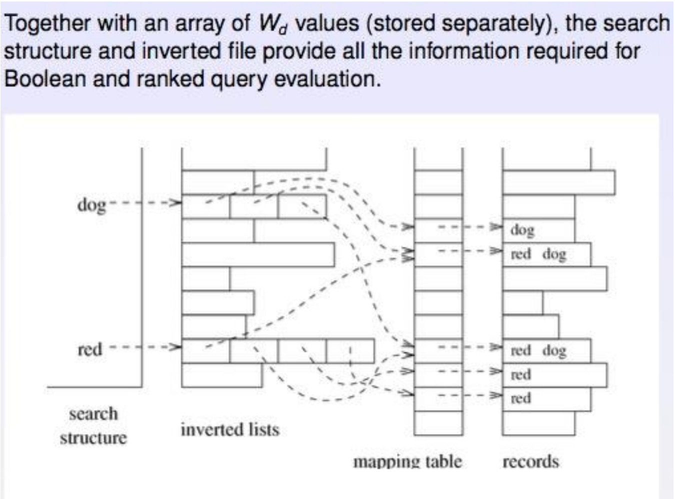
- A pointer to the start of the corresponding inverted list.
- A count f_t of the documents containing t.

That is, the search structure contains the *vocabulary*.

Inverted lists

For each distinct word t, the inverted list contains:

- The identifiers d of documents containing t, as ordinal numbers.
- The associated frequency $f_{d,t}$ of t in d. (We could instead store $w_{d,t}$ or $w_{d,t}/W_d$.)



Together with an array of W_d values (stored separately), the search structure and inverted file provide all the information required for Boolean and ranked query evaluation.

5. Querying

- Query using an inverted file (每次从最短 query list 开始消除(因为可以消除最多的 files))
  - Accumulator costs:
    1) With the standard query evaluation algorithm and long queries, most accumulators are non-zero and an array is the most space and time-efficient structure.
    2) The accumulators are required on a per-query basis.
    3) If only low ft, that is rare terms are allowed to create accumulators, the number of accumulators is greatly reduced.
    4) A simple mechanism is to impose a limit L on the number of accumulators.
  - Querying costs: (Approximations can be used to reduce querying costs, which can affect the answer set in unpredictable ways)
    1) Disk space: for the index
    2) Memory space: for accumulators, vocabulary and caching of

previous results

3) CPU time: for processing inverted lists and updating accumulators
4) Disk traffic: to fetch inverted lists

6. Link analysis

In general search, each document in considered independently.

In web search, a strong piece of evidence for a page’s importance is given by links, in particular how many other pages have links to this page.

- PageRank:
  - ➢ Basic intuition of PageRank:
    1) 如果一个网页被很多其他网页链接到的话，说明这个网页比较重要，也就是 PageRank 值会相对较高。(入度影响自己)
    2) 如果一个 PageRank 值很高的网页链接到一个其他的网页，那么被链接到的网页的 PageRank 值会相应的提高。(出度影响别人)

Machine Learning

Chapter6 Introduction

Chapter7 Introduction to basic probability

1. Probability

- The Axioms of Probability:
  - ➢ A and B are mutually exclusive (互相排斥): P(A or B) = 1
  - ➢ A and B are not mutually exclusive: P(A or B) = P(A) + P(B) – P(A and B)
- Conditional Probability:
  - ➢ P(A|B) = P(A and B) / P(B)
  - ➢ Product rule: P(A and B) = P(A|B) * P(B)
  - ➢ Sum rule: P(A) =$\sum_B P(A\ and\ B)\ = \sum_B P(A|B) * P(B)$
  - ➢ Bayes Theorem: P(B/A) = P(A and B) / P(A) = P(A|B) *P(B)/P(A)= P(A|B) P(B)/$\sum_{\boldsymbol{B}} \boldsymbol{P}(\boldsymbol{A}|\boldsymbol{B})P(B)$
  - ➢ P(A) + P(~A) = 1
  - ➢ P(A) = P(A|B) + P(A|~B)
  - ➢ P(A|B) + P(A|~B) ~= 1
  - ➢ P(A|B) + P(~A|B) = 1

2. Probabilistic Graphical Models (PGM)

- joint distribution (联合分布): p(x,y,z) = p(x)p(y,z|x) = p(x)p(y|x)p(z|x,y)
- Directed Acyclic Graphs (会出题！！！！)
  - ➢ P(L,S) = P(L)P(S)
  - ➢ P(L,S|I) $\neq$ P(L|I) P(S|I)

Chapter8 Data Mining

1. What is Data Mining?

- Databases, Data Mining, Machine Learning and Statistic and their relationships:

  Data mining exploits concepts from db (scalability,semantics), AI(heuristics), Statistics (significance analysis, distributions, etc), Maching learning(various clustering, classification,regression,etc) and Pattern mining(image processing)

- Data mining's goal:
  - ➢ to handle large, high dimensional and complex data (relational, time series, graphical, text, etc)
  - ➢ to discover interesting patterns that can help understand underlying process
  - ➢ to discover knowledge that is useful/actionable
- Data mining tasks involve:
  - ➢ Prediction Methods (supervised learning methods) 需要训练集
    - ✓ Use some variables to predict unknown or future values of other variables. (Classification/Regression/Deviation Detection)
  - ➢ Description Methods (unsupervised learning) 不需要训练集
    - ✓ Find human-interpretable patterns that describe the data (Clustering/Association Rule Discovery/Sequential pattern Discovery)

	Supervised learning	Unsupervised learning
Continuous	**Regression** For example, predict stock price	**PCA** For example, given 1 million data objects with 1000 features, find the most 50 strong features
Categorical	**Classification** For example, classify whether an email is spam or not	**Clustering** For example, gene expression clustering

2. Classification

- What is classification?
  - ➢ Given a collection of training data: Each item of the data contains a set of attributes (features), at least one of the attributes is the class
  - ➢ The goal is to model for class attributes as a function of the values of other attributes.
  - ➢ The discovered model is used to predict the label of a previously unseen item. (for determining how good the discovered model is a test set. Usually, the given data set is partitioned into two disjoint training and test sets. The training set is used to build the model and the test validates it.)

3. Regression

- What is regression?
  - ➢ Predict a value of given continuous valued variable based on the values of other variables, assuming a linear or nonlinear model of dependency.
  - ➢ Great studied in statistics and neural network fields

4. Deviation/Anomaly Detection
   - What is deviation detection?
     - ➢ detect significant deviations from normal behavior

5. Clustering
   - What is clustering?
     - ➢ Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
       - ✓ Data points in one cluster are more similar to one another (need to define suitable similarity function for the application)
       - ✓ Data points in separate cluster are less similar to one another
   - Similarity Measures:
     - ➢ Euclidean Distance: if attributes are continuous
     - ➢ Cosine measure

6. Association Rule Discovery
   - What is association rule discovery?
     - ➢ Given a set of records each of which contain some number of items from a given collection
     - ➢ Produce dependency rules, which will predict occurrence of an item based on occurrences of other items.

7. Sequential Pattern Discovery
   - What is sequential pattern discovery?
     - ➢ Given a set of objects, with each object associated with its own timeline of events, find rules that predict strong sequential dependencies among different events.

Chapter9 Introduction Classification

1. Naïve Bayesian classifier (assumption: attributes are conditionally independent)
   - Formula:

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})} \qquad \left(posteriori = \frac{likelihood \times prior}{evidence} \right)$$

   - Conclusion
     - ➢ NB is simple to build, extremely fast to make decision, and easy to change the probabilities when the new data becomes available (especially when the new data is additional information not modification to previously used data)
     - ➢ Works well in many application areas
     - ➢ Scales easily for large number of dimensions and data sizes
     - ➢ Easy to explain the reason for the decision made

2. Evaluate Classifier
   - For two classes problem:
     - ➢ A classifier may classify a Positive instance as Positive (TP) or as Negative (FN). A Negative instance as Positive (FP) or as Negative (TN)
     - ➢ TP+FN 表示实例的正样本数，FP+TN 表示实例的负样本数

- P/N 表示分类器分类结果， T/F 表示实际的类
- Accuracy = (TP+TN) / (TP+TN+FP+FN) (主对角线/全部值)
- Sensitivity = TP/(TP+FN)
- Specificity = TN/(TN+FP)
- Recall = TP/(TP+FN)
- Precision = TP/ (TP+FP)
- F1_Score = 2 Recall * Precision/(Recall+Precision)

- Evaluation schemes:
  - Leave-One-Out: (N-1 个样本做训练,1 个样本做测试,如此从 N1 到 Nn 本里所有对象都经历过测试和训练,用这个结果的平均值来衡量模型的性能)
    - ✓ Assume we have N data points for which we know the labels. We choose each data point as test case and the rest as training data.
    - ✓ This means we have to train the system N times and the average performance is computed

    - ✓ Good points: 1) there is no sampling bias in evaluating the system and the results will be unique and repeatable for a given method. 2) Generate higher accuracy.
    - ✓ Bad points: not scale well if we have large data set and the training is itself very expensive

  - 10 Fold cross validation:
    - ✓ Assume we have N data points for which we know the labels. We partition the data into 10 (approximately) equal size partitions. We choose each partition for testing and the remaining 9 partitions for testing.
    - ✓ This means we have to train the system 10 times and the average performance is computed.

    - ✓ Good points: only need to train 10 times. (Suitable for large data set)
    - ✓ Bad points: 1) There can be a bias in evaluating the system due to sampling. 2) The results will not be unique unless we always partition the data identically. Generate lower accuracy.

Chapter10 Decision tree

1. Issues:
   - How to build optimal DT for a given training data set?
     Optimal construction of a Decision Tree is NP hard.
   - How to choose attribute values at each decision point?
     Choose an attribute to partition the data at the node such that each partition is as homogenous (least impure) as possible.
   - How to choose number of branches at each node and attribute values for partitioning the data?
     Most of the instance belonging to as few classes as possible and each

partition should be as large as possible. (partition 的结果可以将大部分 Instance 归属到一小部分的 class 下)

- When to stop the growth of the tree?
  Stop the growth of the tree if all the leaf nodes are largely dominated by a single class.

2. Measures of Node Impurity (均是值越大表示越不纯)

- Gini Index:

$$GINI(t) = 1 - \sum_j [p(j\,|\,t)]^2$$

➢ Maximum value of Gini index = (1-1/nc) when records are equally distributed among all classes, implying least interesting or most impure

➢ Minimum is 0 when all records belong to one class, implying most interesting information or most pure or most homogeneous

➢ GINI split:

Used in CART, SLIQ, SPRINT.

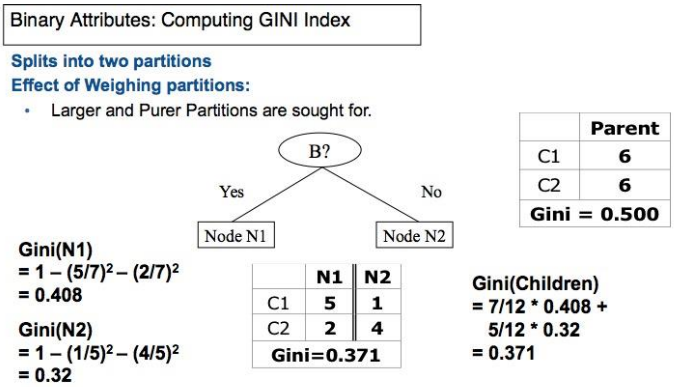
When a node p is split into k partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i,

n = number of records at parent node p.

If GINI(j) – GINI $_{split}$(j) > delta then split the node j.



Binary Attributes: Computing GINI Index

Splits into two partitions

Effect of Weighing partitions:

- Larger and Purer Partitions are sought for.

	Parent
C1	6
C2	6
Gini = 0.500	

Gini(N1)
= 1 – (5/7)² – (2/7)²
= 0.408

Gini(N2)
= 1 – (1/5)² – (4/5)²
= 0.32

	N1	N2
C1	5	1
C2	2	4
Gini=0.371		

Gini(Children)
= 7/12 * 0.408 +
5/12 * 0.32
= 0.371

- Entropy:

$$Entropy(t) = -\sum_j p(j\,|\,t)\log p(j\,|\,t)$$

➢ Maximum (log nc) when records are equally distributed among all classes implying least information

➢ Minimum 0 when all records belong to one class, implying most information

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure. (比如说以 ID 划分，这样的划分效果并没有用， 所有 instance 被分到不同的 class 中)
- Gain Ratio (Overcome the disadvantages of IG):

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

Adjust IG by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized.

- Misclassification error:

$$Error(t) = 1 - \max_{i} P(i \mid t)$$

  - Maximum (1-1/nc) when records are equally distributed among all classes, implying least interesting information.
  - Minimum (0) when all records belong to one class, implying most interesting information.

- Comparison among Splitting Criteria: Entropy>Gini>Misclassification error

3. Three Induction(归纳):
   - Greedy strategy: Choose the attribute and splitting test to partition the records that optimizes certain criterion.
   - Stopping Criteria
   - Decision making: Traverse the branch of decision tree from the root node matching the corresponding attributes values of the test record and the traversal reaches the Leaf Node. Make the decision based on the Leaf node Class distribution
   - Advantages: (代价不高+正确率可以接受)
     - Inexpensive to construct
     - Extremely fast at classifying unknown records
     - Easy to interpret for small-size trees
     - Accuracy is comparable to other classification techniques for many simple data sets.

Chapter11 Rule Based Classification and Random Forest

1. Rule based classification
   - One good approach: Construct a decision tree and Extract one rule for each leaf node.
     - Advantages:
       1) All rules are mutually exclusive this implies only one rule will be applicable at the time of decision making
       2) Easy to extract the rules from traversing the decision tree.
     - Disadvantages:

1) Too restrictive in terms of number rules learned
2) The rules may be too specific and can be very complex

2. Variants of Decision Trees

- Decision tree issue: Decision Trees cannot capture full information available in the data for data sets with large number of dimensions.
- Solutions:
  - ➢ Bagging: (生成 N 课树，减少单决策树带来的毛病) Building several Decision Trees by sampling the data.
    - ✓ For a data set containing N records, we sample N records randomly with replacement.
    - ✓ On average each sample will have (2/3) of N original records
    - ✓ For each sample we build a decision Tree.
    - ✓ We may build 20-50 such trees
    - ✓ At the time of classification we collect all the decisions from the decision trees and apply majority rule to make the final decision.

  - ➢ Random Forests:
    - ✓ At each node we firstly randomly select "m" features out of all the features
    - ✓ There is no need to prune (to reduce the size of the tree) the constructed trees due to the randomness imposed both on the selection of data and the tree construction (因为选择数据以及构建树的随机性质，不需要修剪树-→ already limit the number of features and the size of tree)
    - ✓ Random forests can make use of large number attributes unlike standard decision trees
    - ✓ ---
    - ✓ Advantages:
      1) The best performance is obtained for Random Forests when the trees are not correlated to each other (largely independent of each other). That is the overlap of splitting attributes between any two trees is small.
      2) Able to make the tree random and less correlated
      3) The quality of a decision tree depends on its accuracy which means large m value. (m 值越大树的质量越好)
      4) The independence of each tree depends on the small values of m (m 越小树与树之间的差异性会越大，树的独立性越强说明决策结果的相关性会越弱，系统总体的正确率会越高)
      5) Overall quality of the Random Forest depends on both quality of the trees and their dependence. (说明 m 值需要利用实验来测，trade off 单棵树的 quality 和 树与树之间的独立性)

3. Practical Issues of Classification

- Underfitting and Overfitting: For decision tree we need to make sure it does not over fit the data. One way to do this is to have a stopping criterion that makes sure that if GINI index is smaller than some threshold value we stop splitting the node. (improve accuracy)

- Missing Values: Missing values can be predicated (断言) by using the distribution of the attribute values. Especially if we find two highly correlated attributes we can use this correlation to predict the missing value.

- Costs of Classification: we need to optimize the decision based on cost involved using confusion matrix. (Minimize the function below)

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{Cost}_{ij} f_{ij}$$

Chapter12 KNN and Support Vector Machines

1. K-Nearest Neighbour (KNN)

- Geometric (几何学) based method
- Simple technique and works well in many situations
- No learning involved
- Given a training data set with labelled information and test case we find K training data points nearest to the test case.
- The test case is labelled with the label of the most frequent label of the K nearest points to the test case.
- Advantages:
  - ➢ Very simple scheme with no learning phase
  - ➢ Works well for small dimensional data
  - ➢ New data can be easily added and hence it is a Lazy Method.
  - ➢ No need to estimate any probabilities but the notion of K nearest implicitly the distribution when determining the K-nearest points (try many k value to find which is better)

- What should be the distance function?

$$\text{Euclidian distance } (p,q) = \|p-q\|^2 = \sqrt{\sum_{i=1}^{n}(pi-qi)^2}$$

$$\text{Minkowski distance}(p,q,k) = \sqrt[k]{\sum_{i=1}^{n}|\,pi-qi\,|^k}$$

$$Radial\_Basis\_dist(\mathbf{p},\mathbf{q},\sigma) = \exp(-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma^2})$$

2. Support Vector Machines

- SVM is a statistic machine learning technique (like Naïve Bayes) and is a geometric based method (like KNN).
- Separating Hyper-plane: Unfortunately, we may not find hyper-panes that can separate given set of training instances. Therefore we allow some margin of errors.
  - ➢ Lagrange function
  - ➢ Kernel mapping (may construct a better optimal separating hyper-plane into a higher dimensional feature space)
- Weaknesses of SVM
  - ➢ Best performance depends on the choice of kernel and its parameters
  - ➢ Learning is very expensive. Generalization to multi-class memberships.

- Solution:
  - ➢ Build one versus the rest
  - ➢ Build one versus one
  - ➢ Build log(n) binary classifiers

Chapter13 Clustering

1. Definition of Clustering: Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the other objects in other groups.

2. Cluster Analysis: Understanding/Summarization
   - What is not Cluster Analysis?
     - ➢ Supervised classification (Have class label information)
     - ➢ Simple segmentation (Dividing students into different registration groups alphabetically, by last name)
     - ➢ Results of a query (Groupings are a result of an external specification)
     - ➢ Graph partitioning (Some mutual relevance and synergy(协同),but areas are not identical)

3. Types of Clustering
   - A clustering is a set of clusters
   - Partitional Clustering (K-means): A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.
   - Hierarchical clustering (有点类似 Huffman 算法): A set of nested(嵌入的) clusters organized as a hierarchical tree.
   - Other distinction between set of clusters:
     - ➢ Exclusive(专一的) vs. non-exclusive: In non-exclusive clustering, points may belong to multiple clusters (Can represent multiple classes or ‘border’ points)
     - ➢ Fuzzy vs. non-fuzzy: In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1.
     - ➢ Partial vs. Complete: In some cases, we only want to cluster some of the data.
     - ➢ Heterogeneous vs. Homogenous: Cluster of widely different sizes, shapes, and densities.

4. Types of Clusters
   - ➢ Well-Separated Clusters (easiest clustering): A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.
   - ➢ Center-based: A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster. (The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most representative point of a cluster) ------→ Tutorial question
   - ➢ Contiguous Cluster (Nearest neighbour or Transitive): A cluster is a set of

points such that a point in a cluster is closer to one or more other points in the cluster than to any point not in the cluster.

- Density-based: A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density. (Used when the clusters are irregular or intertwined (错综复杂的), and when noise and outliers are present)
- Shared Property or Conceptual Clusters: Finds clusters that share common property or represent a particular concept.

5. Cluster Algorithms

- K-means and its variants:
  - ✓ Partitional clustering approach
  - ✓ Each cluster is associated with a centroid (center point)
  - ✓ Each point is assigned to the cluster with the closest centroid
  - ✓ Number of clusters K, must be specified
  - ✓ Initial centroids are often chosen randomly
  - ✓ The centroid is the mean of the points in the cluster
  - ✓ "Closeness" is measured by Euclidean distance, cosine similarity, correlation etc.
  - ✓ Most of the convergence (收敛) happens in the first few iterations
  - ✓ Evaluating K-means Clusters (Most common measure is Sum of Squared Error (SSE) 平方误差和)One easy way to reduce SSE is to increase K, the number of clusters (A good clustering with smaller K can have a lower SSE than a poor clustering with higher K)
  - ✓ Solutions to Initial Centroids Problem:

    Multiple runs/Sample and use hierarchical clustering to determine initial centroids/Select more than k initial centroids and then select among these initial centroids/Postprocessing/Bisecting K-means
  - ✓ Handling Empty Cluster:

    Basic K-means algorithm can yield empty clusters

    Several strategies:

    1) Choose the point that contributes most to SSE
    2) Choose a point from the cluster with the highest SSE
    3) If there are several empty cluster, the above can be repeated several times.
  - ✓ Pre-processing and Post-processing:
    - Pre-processing:

      Normalize the data/Eliminate outliers
    - Post-processing

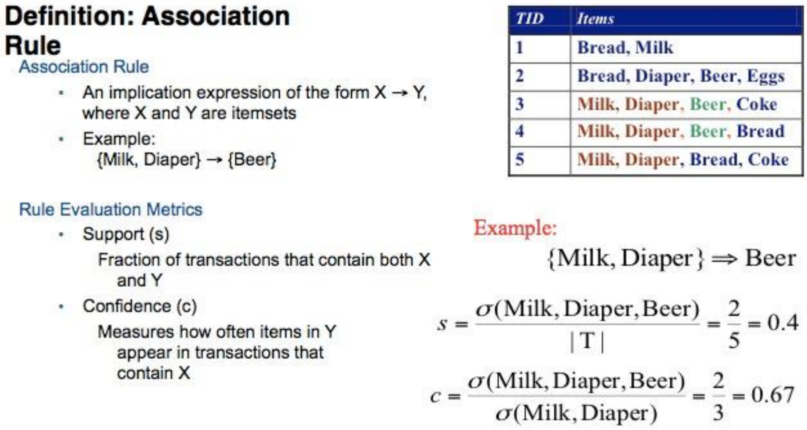
      Eliminate small clusters that may represent outliers/Split 'loose' clusters/Merge clusters that are 'close' and that have relatively low SSE

- Hierarchical clustering
- Density-based clustering

Chapter14 Data Mining Association Analysis

1. Association Rule Mining

- Definition:
  - ➢ Association Rules: Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.
  - ➢ Frequent Itemset: A collection of one or more items. (k-itemset: An Itemset that contains k items)
  - ➢ Support: Fraction of transactions that contain an itemset
  - ➢ Frequent itemset: An itemset whose support is greater than or equal to a minsup threshold.



Definition: Association Rule

Association Rule

- An implication expression of the form X → Y, where X and Y are itemsets
- Example:
  {Milk, Diaper} → {Beer}

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Milk, Diaper, Beer, Bread
5	Milk, Diaper, Bread, Coke

Rule Evaluation Metrics

- Support (s)
  Fraction of transactions that contain both X and Y
- Confidence (c)
  Measures how often items in Y appear in transactions that contain X

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

- Association Rule Mining Task

  Given a set of transactions T, the goal of association rule mining is to find all rules having: support >= minsup threshold, confidence >= minconf threshold

  - ➢ Brute force approach: (关联规则所花费的时间主要是在生成频繁集上，因为找出的频繁集往往不会很多，利用频繁集生成规则也不会花太多时间，而生成频繁集需要测试很多备选项集，O(2^n))
    - ✓ List all possible association rules
    - ✓ Compute the support and confidence for each rule
    - ✓ Prune rules that fail the minsup and minconf threshold

- Mining Association Rule

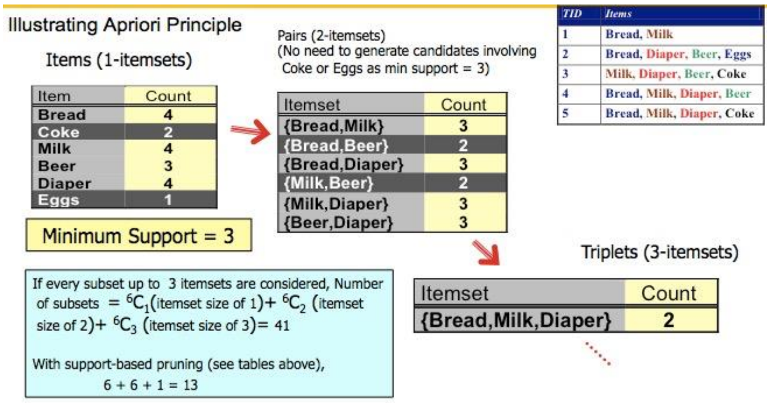
  Rules originating from the same itemset have identical support but can have different confidence

  Two-step approach:

  1) Frequent Itemset Generation: Generate all itemsets whose support >= minsup
  2) Rule Generation: Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

- Frequent Itemset Generation Strategies
  - ➢ Reduce the number of candidates: Use pruning techniques to reduce (如果一个集合不是频繁项集，则它的所有超集都不是频繁项集)

    Apriori principle: if an itemset is frequent, then all of its subsets must also be frequent



Illustrating Apriori Principle

Items (1-itemsets)

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Minimum Support = 3

Pairs (2-itemsets)
(No need to generate candidates involving Coke or Eggs as min support = 3)

Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	2

If every subset up to 3 itemsets are considered, Number of subsets = ${}^6C_1$(itemset size of 1)+ ${}^6C_2$ (itemset size of 2)+ ${}^6C_3$ (itemset size of 3)= 41

With support-based pruning (see tables above),
6 + 6 + 1 = 13

- ➢ Reduce the number of transactions: Used by DHP (Direct Hashing and Pruning) and vertical-based mining algorithms
- ➢ Reduce the number of comparisons: No need to match every candidate against every transaction

  To reduce the number of comparisons, store the candidates in a hash structure (instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets)

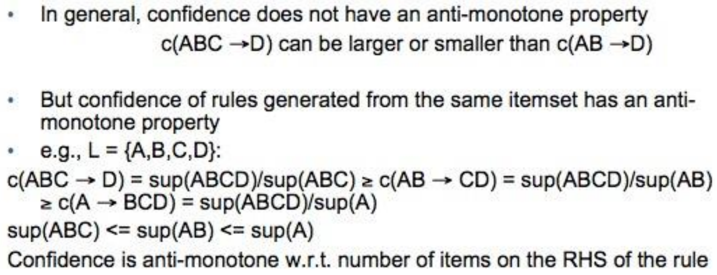
  - ✓ Generate Hash Tree
  - ✓ Subset Operation Using Hash Tree (进行一次 hash tree 的遍历就可以完成对该 transaction 的计数)

- Rule Generation

  Given a frequent itemset L, find all non-empty subsets F⊂L such that F->L-F satisfies the minimum confidence requirement

  How to efficiently generate rules from frequent itemsets?

  Confidence of rules generated from the same itemset has an anti-monotone (反单调)property.



- In general, confidence does not have an anti-monotone property
  c(ABC →D) can be larger or smaller than c(AB →D)

- But confidence of rules generated from the same itemset has an anti-monotone property
- e.g., L = {A,B,C,D}:

c(ABC → D) = sup(ABCD)/sup(ABC) ≥ c(AB → CD) = sup(ABCD)/sup(AB)
≥ c(A → BCD) = sup(ABCD)/sup(A)
sup(ABC) <= sup(AB) <= sup(A)
Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

因此可以在 generate rules 的时候进行修剪(prune) 以及合并(join)